โมเดลการเรียนรู้แบบไม่มีผู้สอนสำหรับการตรวจจับความผิดปกติแบบเรียลไทม์ในระบบเครือข่ายคอมพิวเตอร์

เกรียงไกร ลิ่มทอง

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยกรุงเทพ
Email: kriangkrai.l@bu.ac.th

บทคัดย่อ

ความผิดปกติในระบบเครือข่ายคอมพิวเตอร์เป็นปัญหาสำคัญที่นักวิจัยจำนวนมากให้ความสนใจ จึงได้มีการนำเสนอวิธีตรวจจับรูปแบบต่างๆ ในช่วงระยะเวลากว่าสิบปีที่ผ่านมา แต่ส่วนใหญ่แล้ววิธีดังกล่าวจำเป็นต้องรวบรวมข้อมูลทั้งหมดก่อนที่จะนำไปประมวลผล ซึ่งวิธีการเช่นนี้ไม่สามารถนำไปใช้กับการตรวจจับแบบเรียลไทม์ได้ บทความนี้จึงได้นำเสนอโมเดลการเรียนรู้แบบไม่มีผู้สอนสำหรับการตรวจจับความผิดปกติแบบเรียลไทม์ในระบบเครือข่ายคอมพิวเตอร์ ผลการทดลองแสดงให้เห็นว่าโมเดลที่นำเสนอสามารถตรวจจับความผิดปกติได้หลายประเภทและมีประสิทธิภาพ อีกทั้งยังมีความยืดหยุ่นในการประยุกต์ใช้งาน เพราะสามารถเลือกใช้อัลกอริทึมหรือเลือกใช้คุณลักษณะของข้อมูลจราจรทางคอมพิวเตอร์ได้หลายรูปแบบ นอกจากนี้เวลาในการประมวลผลที่ได้จากการทดลองช่วยยืนยันว่าโมเดลที่นำเสนอสามารถตรวจจับความผิดปกติแบบเรียลไทม์ได้

คำสำคัญ-- ระบบเรียลไทม์; การตรวจจับความผิดปกติ; การเรียนรู้แบบไม่มีผู้สอน; ข้อมูลจราจรทางคอมพิวเตอร์

1. บทนำ

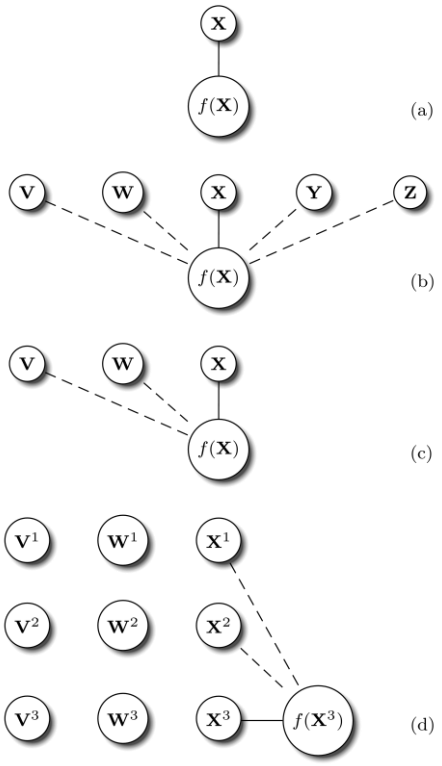
ข้อมูลจราจรทางคอมพิวเตอร์ทั้งในระบบเครือข่ายอินเตอร์เน็ตและระบบเครือข่ายภายในมีปริมาณเพิ่มขึ้นเป็นจำนวนมากในช่วงระยะเวลาไม่กี่ปีที่ผ่านมา จึงเป็นการยากที่ผู้ดูแลระบบจะเข้าไปทำการตรวจสอบแต่ละชุดข้อมูลที่มีการรับส่งผ่านระบบเครือข่าย นอกจากนี้การโจมตีของระบบคอมพิวเตอร์ที่เกิดขึ้นในปัจจุบัน มีการใช้เทคนิคที่มีความซับซ้อนมากขึ้นเพื่อหลบหลีกระบบตรวจจับการบุกรุก ซึ่งโปรแกรมที่ใช้โจมตีดังกล่าวมีอยู่แพร่หลายและหามาใช้ได้โดยง่าย อีกทั้งยังมีความผิดปกติที่เกิดขึ้นจากสาเหตุหรือปัจจัยภายใน อาทิเช่น เกิดจากอุปกรณ์เสียหาย, เกิดจากการเปลี่ยนแปลงโครงสร้างระบบเครือข่าย, หรือเกิดจากผู้ใช้งานจำนวนมากกว่าการใช้งานปกติ เป็นต้น ความผิดปกติทั้งหลายเหล่านี้มีผลกระทบทั้งทางตรงและทางอ้อมต่อความมั่นคงและความน่าเชื่อถือของระบบเครือข่าย ดังนั้นการบริหารระบบเครือข่ายคอมพิวเตอร์ในปัจจุบันจึงมีความต้องการระบบอัตโนมัติสำหรับตรวจจับความผิดปกติหลากหลายรูปแบบ ทั้งที่เกิดจากการบุกรุกและเกิดจากเหตุบังเอิญ

จากการศึกษางานวิจัยที่ผ่านมา [1,2] พบว่ารูปแบบการตรวจจับความผิดปกติในระบบเครือข่ายคอมพิวเตอร์สามารถแบ่งออกได้เป็น 2 ประเภทตามวิธีการตรวจจับ คือ วิธีตรวจจับโดยการใช้ลายเซ็นและวิธีตรวจจับโดยการใช้สถิติ

วิธีตรวจจับโดยการใช้ลายเซ็น (signature-based technique) เป็นการเปรียบเทียบข้อมูลจราจรคอมพิวเตอร์กับฐานข้อมูลความผิดปกติ ซึ่งฐานข้อมูลดังกล่าวจะถูกปรับปรุงให้ทันสมัยโดยผู้เชี่ยวชาญ วิธีการเช่นนี้มีความถูกต้องแม่นยำสูงและสามารถเริ่มใช้งานได้ทันทีหลังจากติดตั้ง แต่ไม่สามารถตรวจจับความผิดปกติรูปแบบใหม่ที่ยังไม่ถูกระบุอยู่ในฐานข้อมูลได้ อีกวิธีหนึ่ง วิธีตรวจจับโดยการใช้สถิติ (statistical-based technique) เป็นการเปรียบเทียบข้อมูลจราจรคอมพิวเตอร์โดยใช้หลักการทางสถิติ ซึ่งมีสมมุติฐานว่าข้อมูลส่วนใหญ่เป็นข้อมูลปกติ ส่วนข้อมูลที่แตกต่างจากข้อมูลส่วนใหญ่เป็นข้อมูลผิดปกติ วิธีการเช่นนี้สามารถตรวจจับความผิดปกติรูปแบบใหม่ได้ แต่การตรวจจับด้วยวิธีนี้จำเป็นต้องเรียนรู้และจดจำข้อมูลปกติในช่วงระยะเวลาหนึ่งก่อน ซึ่งทำให้ไม่สามารถเริ่มใช้งานได้ทันทีหลังจากติดตั้ง ด้วยเหตุนี้จึงทำให้งานวิจัยส่วนใหญ่มุ่งพัฒนาวิธีตรวจจับโดยการใช้สถิติ เพื่อให้มีความถูกต้องแม่นยำมากขึ้น และลดการเรียนรู้ให้มีระยะเวลาที่สั้นลง

หลังจากศึกษางานวิจัยเกี่ยวกับวิธีตรวจจับโดยการใช้สถิติพบว่างานวิจัยส่วนใหญ่ใช้วิธีการประมวลผลข้อมูลย้อนหลัง หลังจากรวบรวมข้อมูลเป็นช่วงระยะเวลาหนึ่ง (batch processing) ซึ่งวิธีนี้ไม่สามารถนำไปใช้ได้กับการประมวลผลแบบเรียลไทม์ (real-time processing) นอกจากนี้ในขั้นตอนการเรียนรู้ข้อมูลหลังจากติดตั้งระบบ เป็นไปได้ยากที่จะคัดแยกข้อมูลเพื่อบอกให้กับระบบว่าข้อมูลไหนเป็นข้อมูลปกติหรือข้อมูลไหนเป็นข้อมูลผิดปกติของระบบเครือข่าย ซึ่งข้อมูลดังกล่าวมีผลกระทบอย่างมากในขั้นตอนการตรวจจับความผิดปกติ

งานวิจัยนี้จึงได้นำเสนอโมเดลการเรียนรู้แบบไม่มีผู้สอนสำหรับการตรวจจับความผิดปกติแบบเรียลไทม์ในระบบเครือข่ายคอมพิวเตอร์ ซึ่งโมเดลดังกล่าวไม่จำเป็นต้องคัดแยกข้อมูลในขั้นตอนการเรียนรู้หลังจากติดตั้งระบบ นอกจากนี้ยังสามารถเลือกใช้อัลกอริทึมในการเรียนรู้ (learning algorithm) หรือเลือกใช้คุณลักษณะของข้อมูล (feature) โดยไม่มีข้อจำกัดว่าต้องเป็นอันหนึ่งอันใดโดยเฉพาะ ซึ่งทำให้ยืดหยุ่นต่อการนำไปใช้งานในสภาวะระบบเครือข่ายหรือโปรโตคอลที่ต่างกันได้



รูปที่ 1. โมเดลการเรียนรู้ (a) แบบใช้ผู้เชี่ยวชาญ, (b) แบบเชิงกลุ่ม, (c) แบบเรียลไทม์, และ (d) แบบที่นำเสนอ

2. งานวิจัยที่เกี่ยวข้องและโมเดลที่นำเสนอ

เพื่อให้เข้าใจง่ายต่อการอธิบาย สมมุติให้มีเพียงหนึ่งชุดข้อมูลเกิดขึ้นทุกหนึ่งวินาทีบนแกนเวลาที่มีความยาวหนึ่งวัน ข้อมูล **X** คือชุดข้อมูลที่เกิดขึ้น ณ เวลาที่ t = X บนแกนเวลาปัจจุบัน แต่ถ้าอธิบายถึงชุดข้อมูลที่เกิดขึ้นบนแกนเวลามากกว่าหนึ่งแกน ข้อมูล $\mathbf{X}^i$ คือชุดข้อมูลที่เกิดขึ้น ณ เวลาที่ t = X บนแกนเวลา i ดังนั้นเป้าหมายของการตรวจจับความผิดปกติคือการหาฟังก์ชัน f(**X**) สำหรับการคัดแยกข้อมูล (discriminant function) ว่าชุดข้อมูล **X** เป็นข้อมูลปกติหรือเป็นข้อมูลผิดปกติ

วิธีการที่ง่ายที่สุดในการสร้างฟังก์ชันคัดแยกข้อมูลดังกล่าวคือให้ผู้เชี่ยวชาญเป็นคนกำหนดฟังก์ชัน f(**X**) เรียกวิธีนี้ว่าเป็นโมเดลการเรียนรู้แบบใช้ผู้เชี่ยวชาญ ดังแสดงในรูปที่ 1-(a) จากรูปได้แสดงความสัมพันธ์ (เส้นทึบ) ระหว่าง **X** ชุดข้อมูลที่ต้องการคัดแยกและ f(**X**) ฟังก์ชันคัดแยกข้อมูล โมเดลการเรียนรู้แบบนี้อาจเปรียบได้กับวิธีตรวจจับโดยการใช้ลายเซ็น (signature-based technique) เพราะใช้ผู้เชี่ยวชาญระบุความผิดปกติและเก็บไว้ในฐานข้อมูลเช่นกัน โมเดลการเรียนรู้แบบใช้ผู้เชี่ยวชาญนี้ถูกนำไปประยุกต์ใช้ในโปรแกรมป้องกันไวรัสคอมพิวเตอร์, ไฟร์วอลล์, และอุปกรณ์ตรวจจับการบุกรุกโดยการใช้ลายเซ็น เช่น Snort [3] เป็นต้น ข้อเด่นของโมเดลนี้คือมีความถูกต้องแม่นยำสูง สามารถนำไปใช้ตรวจจับความผิดปกติแบบเรียลไทม์ได้ สามารถเริ่มใช้งานได้ทันทีหลังจากติดตั้ง แต่ข้อด้อยคือไม่ยืดหยุ่นต่อการทำงาน ตรวจจับได้เฉพาะความผิดปกติที่ผู้เชี่ยวชาญระบุอยู่ในฐานข้อมูล ไม่สามารถตรวจจับความผิดปกติรูปแบบใหม่ได้

เพื่อสร้างฟังก์ชันคัดแยกข้อมูลโดยอัตโนมัติ นักวิจัยได้ศึกษาและนำเสนอวิธีการที่เราเรียกว่าโมเดลการเรียนรู้แบบเชิงกลุ่ม [4] ดังแสดงในรูปที่ 1-(b) จากรูป นอกจากแสดงความสัมพันธ์ของ **X** ชุดข้อมูลที่ต้องการคัดแยกแล้ว (เส้นทึบ) ยังได้แสดงถึงความสัมพันธ์ระหว่างชุดข้อมูลที่ใช้สร้างฟังก์ชันคัดแยกข้อมูลโดยอัตโนมัติ (เส้นประ) ตัวอย่างจากรูป f(**X**) ถูกสร้างขึ้นโดยอัตโนมัติจากชุดข้อมูล **V**, **W**, **Y**, และ **Z** หลังจากนั้น f(**X**) จึงถูกนำไปใช้ในการคัดแยกชุดข้อมูล **X** โมเดลการเรียนรู้แบบนี้อาจเปรียบได้กับวิธีตรวจจับโดยการใช้สถิติที่ทำงานแบบ batch processing เพราะต้องเก็บรวบรวมข้อมูลทั้งหมดแล้วค่อยมาประมวลผลย้อนหลัง ข้อเด่นของโมเดลนี้คือมีความยืดหยุ่นมากขึ้น เพราะฟังก์ชันคัดแยกข้อมูลถูกสร้างขึ้นโดยอัตโนมัติ แต่ข้อด้อยคือมีความถูกต้องแม่นยำต่ำกว่า และไม่สามารถนำไปประยุกต์ใช้ในการตรวจจับความผิดปกติแบบเรียลไทม์ได้ เพราะจำเป็นต้องรวบรวมข้อมูลทั้งหมดก่อนทำการคัดแยก

ถ้าปรับปรุงโมเดลการเรียนรู้แบบเชิงกลุ่มดังแสดงในรูปที่ 1-(b) ให้สร้างฟังก์ชันคัดแยกข้อมูลโดยอัตโนมัติจากชุดข้อมูลก่อนหน้าเท่านั้น เราจะได้โมเดลการเรียนรู้ที่สามารถนำไปประยุกต์ใช้ในการตรวจจับความผิดปกติแบบเรียลไทม์ได้ ดังแสดงในรูปที่ 1-(c) หรือที่เราเรียกว่าโมเดลการเรียนรู้แบบเรียลไทม์ แต่โมเดลนี้มี 2 ประเด็นหลักที่เราต้องพิจารณาคือ (1) กรณีที่ไม่มีชุดข้อมูลก่อนหน้าเราจะสร้างฟังก์ชันคัดแยกข้อมูลได้อย่างไร ตัวอย่างจากรูปที่ 1-(c) ณ เวลาที่ t = V การคัดแยกชุดข้อมูล **V** ซึ่งไม่มีชุดข้อมูลก่อนหน้าจะทำอย่างไร (2) โมเดลการเรียนรู้แบบนี้ผู้โจมตีอาจสร้างชุดข้อมูลหลอกหรือแก้ไขเปลี่ยนแปลงชุดข้อมูลก่อนหน้า เพื่อหลบหลีกการตรวจจับหรือเพื่อส่งผลกระทบให้ฟังก์ชันคัดแยกข้อมูลทำงานไม่ถูกต้องก็เป็นได้

เพื่อแก้ปัญหาของโมเดลแบบเรียลไทม์ เราจึงได้นำเสนอโมเดลการเรียนรู้แบบใหม่ ดังแสดงในรูปที่ 1-(d) โมเดลการเรียนรู้แบบนี้ฟังก์ชันคัดแยกข้อมูลจะถูกสร้างขึ้นโดยอัตโนมัติจากชุดข้อมูลที่เกิดขึ้น ณ เวลาเดียวกันกับชุดข้อมูลที่ต้องการคัดแยก แต่ชุดข้อมูลดังกล่าวเกิดขึ้นบนแกนเวลาก่อนหน้าเท่านั้น ตัวอย่างจากรูปที่ 1-(d) ถ้าต้องการคัดแยกชุดข้อมูล $\mathbf{X}^3$ ซึ่งเป็นชุดข้อมูลที่เกิดขึ้น ณ เวลาที่ t = X บนแกนเวลาที่ 3 (สมมุติว่าเป็นแกนเวลาปัจจุบัน) ฟังก์ชันคัดแยกข้อมูล f($\mathbf{X}^3$) จะถูกสร้างจากชุดข้อมูล $\mathbf{X}^1$ และ $\mathbf{X}^2$ ซึ่งเป็นชุดข้อมูลที่เกิดขึ้น ณ เวลาที่ t = X เช่นเดียวกันกับชุดข้อมูล $\mathbf{X}^3$ แต่ชุดข้อมูล $\mathbf{X}^1$ และ $\mathbf{X}^2$ เกิดขึ้นบนแกนเวลาที่ 1 และแกนเวลาที่ 2 ตามลำดับ ซึ่งเป็นแกนเวลาก่อนหน้าแกนเวลาปัจจุบัน ด้วยวิธีการทำงานเช่นนี้ จึงเป็นการยากที่ผู้โจมตีจะย้อนเวลากลับไปสร้างชุดข้อมูลหลอกหรือแก้ไขเปลี่ยนแปลงชุดข้อมูลก่อนหน้า เพื่อหลบหลีกการตรวจจับหรือเพื่อส่งผลกระทบให้ฟังก์ชันคัดแยกชุดข้อมูลทำงานไม่ถูกต้อง

ดังนั้นเพื่อทดสอบประสิทธิภาพการทำงานเบื้องต้นของโมเดลที่นำเสนอ เราจึงได้ทำการทดลองวัดความถูกต้องแม่นยำในการตรวจจับความผิดปกติ และจับเวลาที่ใช้ในการประมวลผลทั้งในขั้นตอนการเรียนรู้และในขั้นตอนการตรวจจับ โดยประยุกต์ใช้โมเดลที่นำเสนอกับอัลกอริทึมการเรียนรู้ของเครื่อง (machine learning algorithm) รายละเอียดต่างๆเกี่ยวกับการทดลองจะถูกอธิบายในส่วนต่อไป

3. ระเบียบวิธีวิจัยและวิธีการวิจัย

ในส่วนนี้จะอธิบายถึงการรวบรวมข้อมูลสำหรับการทดลอง, คุณลักษณะของข้อมูลจราจรทางคอมพิวเตอร์ที่เลือกใช้ในการทดลอง, อัลกอริทึมการเรียนรู้และการตรวจจับความผิดปกติ, และวิธีวัดประสิทธิภาพการทำงาน

3.1. ข้อมูลสำหรับการทดลอง

ข้อมูลสำหรับการทดลองถูกเก็บรวบรวมมาจากสองแหล่ง ข้อมูลแหล่งที่หนึ่งเป็นข้อมูลจราจรจากศูนย์บริการคอมพิวเตอร์และอินเตอร์เน็ต มหาวิทยาลัยเกษตรศาสตร์ โดยเก็บรวบรวมข้อมูลเป็นระยะเวลา 55 วัน ศูนย์ดังกล่าวมีจำนวนเครื่องคอมพิวเตอร์สำหรับให้บริการประมาณ 175 เครื่อง เปิดบริการระหว่างเวลา 8.00-24.00 น. มีผู้เข้ามาใช้บริการประมาณ 1,300 คนต่อวัน เครื่องคอมพิวเตอร์ทั้งหมดมีการติดตั้งโปรแกรมที่จำเป็นต่อการใช้งานทั่วไป ผู้ใช้บริการไม่สามารถติดตั้งโปรแกรมอื่นเพิ่มเติมได้ ทุกเครื่องติดตั้งโปรแกรมป้องกันไวรัสและมีการปรับปรุงฐานข้อมูลไวรัสให้ทันสมัยอยู่เสมอ จึงอนุมานได้ว่าข้อมูลจราจรทางคอมพิวเตอร์ทั้งหมดเป็นข้อมูลการใช้งานปกติ

ข้อมูลแหล่งที่สองเป็นข้อมูลจราจรที่มีการใช้งานผิดปกติ ถูกจำลองโดยห้องปฏิบัติการวิจัย DARPA Lincoln ตั้งอยู่ที่ Massachusetts Institute of Technology [5] เราได้คัดเลือกเฉพาะข้อมูลจราจรทางคอมพิวเตอร์ที่เป็นข้อมูลผิดปกติ โดยแบ่งออกเป็นประเภทดังต่อไปนี้

1. Back เป็นการโจมตีแบบ denial of service ไปยังพอร์ตหมายเลข 80 ของเครื่องแม่ข่ายที่ติดตั้ง Apache web server
2. IpSweep เป็นเครื่องมือที่ผู้โจมตีใช้ทำการตรวจสอบระบบเครือข่ายเป้าหมายว่ามีการใช้งาน IP address ใดบ้างในระบบ
3. Neptune เป็นการโจมตีแบบ denial of service โดยผู้โจมตีส่ง SYN packet จำนวนมากไปยังหนึ่งพอร์ตหรือหลายพอร์ตของเครื่องเป้าหมาย
4. PortSweep เป็นเครื่องมือที่ผู้โจมตีใช้ทำการตรวจสอบเครื่องเป้าหมายว่ามีการเปิดใช้งานพอร์ตหมายเลขใดบ้าง
5. Smurf เป็นการโจมตีแบบขยาย (amplified attack) โดยใช้หลักการส่งชุดข้อมูล ICMP echo/reply เป็นจำนวนมาก

เราแบ่งข้อมูลการใช้งานปกติออกเป็น 2 ส่วน ส่วนที่หนึ่งจำนวน 39 วัน (≈ 70%) เรียกว่าชุดข้อมูลเรียนรู้ (training data set) ใช้สำหรับอัลกอริทึมในขั้นตอนการเรียนรู้ (learning process) ส่วนที่สองจำนวน 16 วัน (≈ 30%) ถูกนำไปรวมกับข้อมูลการใช้งานผิดปกติในแต่ละประเภท เรียกว่าชุดข้อมูลทดสอบ (test data set) ใช้สำหรับการทดสอบประสิทธิภาพในขั้นตอนการตรวจจับ (detecting process)

3.2. คุณลักษณะของข้อมูลที่เลือกใช้

เรารวบรวมคุณลักษณะของข้อมูลจราจรทางคอมพิวเตอร์ทุกช่วงเวลาหนึ่งวินาที (interval) ดังแสดงในตารางที่ 1 ประกอบด้วยชื่อย่อที่ใช้ในการทดลอง, คุณลักษณะของข้อมูล, และรายละเอียด ในการทดลองเราจะใช้คุณลักษณะของข้อมูลทีละหนึ่งอย่าง เพื่อทดสอบว่าคุณลักษณะของข้อมูลไหนเหมาะสำหรับการตรวจจับความผิดปกติประเภทใด แต่เรายังได้ทำการทดลองโดยใช้คุณลักษณะของข้อมูลพร้อมกันทั้งหมด โดยเราใช้ชื่อย่อว่า f_{all} ซึ่งจะถูกแสดงอยู่ในผลการทดลองเช่นกัน

3.3. อัลกอริทึมการเรียนรู้และการตรวจจับ

เราเลือก 3 อัลกอริทึมการเรียนรู้ของเครื่องเพื่อนำมาใช้ทำงานร่วมกับโมเดลที่นำเสนอ ได้แก่

Multivariate Normal Distribution (MND) เป็นอัลกอริทึมในการหาฟังก์ชันความหนาแน่นของความน่าจะเป็นที่มีการแจกแจงแบบปกติบนพื้นที่ขนาด n มิติ [6] โดย n คือจำนวนคุณลักษณะของข้อมูลที่เลือกมาใช้งาน อัลกอริทึมนี้เหมาะสำหรับข้อมูลที่มีการแจกแจงแบบปกติ โดยในการทดลองนี้เราอนุมานว่าคุณลักษณะของข้อมูลจราจรทางคอมพิวเตอร์ทั้งหมดมีการแจกแจงแบบปกติ

k-Nearest Neighbor (KNN) เป็นอัลกอริทึมอย่างง่ายที่ไม่ต้องการการเรียนรู้ล่วงหน้า [7] แต่จะใช้วิธีการวัดระยะห่างระหว่างข้อมูลทดสอบกับข้อมูลเรียนรู้ทั้งหมดที่ตั้งอยู่บนพื้นที่ขนาด n มิติ โดย n คือจำนวนคุณลักษณะของข้อมูลที่เลือกมาใช้งาน ในการทดลองนี้เราใช้การวัดระยะห่างแบบยุคลิด (Euclidean distance) ซึ่งเป็นมาตรฐานการวัดระยะห่างที่ถูกนำมาใช้กันทั่วไป นอกจากนี้เรากำหนดค่าคงที่ k เท่ากับ 3 ซึ่งเป็นค่าที่นักวิจัยมักเลือกใช้ในการทดลองกับอัลกอริทึมนี้ โดยค่า k มีผลกระทบกับประสิทธิภาพการทำงานของอัลกอริทึม KNN

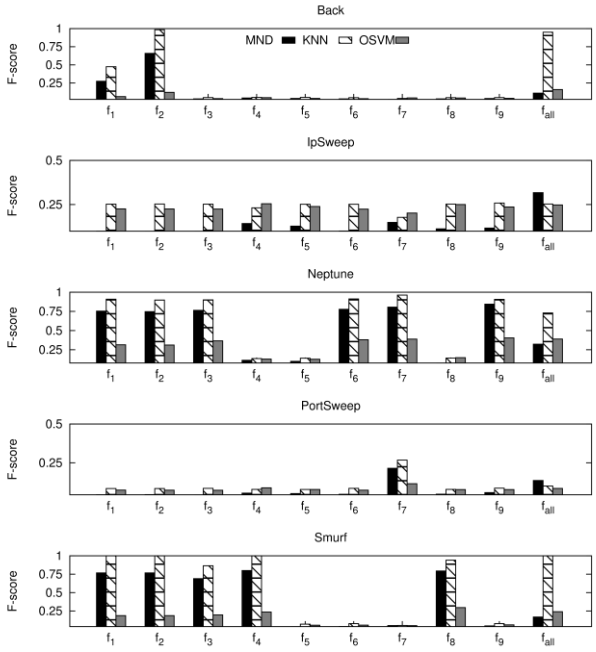
One-Class Support Vector Machine (OSVM) [8] เป็นอัลกอริทึมที่ถูกปรับปรุงมาจากอัลกอริทึม support vector machine ใช้สำหรับการเรียนรู้ชุดข้อมูลที่มีเพียงกลุ่มเดียว (one class) โดยจะทำการเปลี่ยนชุดข้อมูลที่ตั้งอยู่บนพื้นที่มาตรฐานขนาด n มิติให้ไปอยู่ในพื้นที่ใหม่ที่สามารถคัดแยกชุดข้อมูลได้ง่ายกว่าพื้นที่เดิม การเปลี่ยนชุดข้อมูลจากพื้นที่เดิมไปยังพื้นที่ใหม่ดังกล่าวจำเป็นต้องเลือกใช้เคอร์เนลที่เหมาะสมด้วย ในการทดลองเราใช้ชุดคำสั่งของ LIBSVM [9] และใช้เคอร์เนลแบบ radial basis function (RBF)

3.4. วิธีวัดประสิทธิภาพการทำงาน

การวัดประสิทธิภาพในการตรวจจับความผิดปกติ เราใช้ค่า F-score [10] ซึ่งเป็นค่ามาตรฐานสำหรับวัดประสิทธิภาพการคัดแยกข้อมูลที่มีสองกลุ่ม (binary classification) การทดลองในงานวิจัยนี้สามารถคัดแยกชุดข้อมูลออกได้เป็น 2 กลุ่ม คือ กลุ่มข้อมูลปกติและกลุ่มข้อมูลผิดปกติ ซึ่งค่า F-score มีค่าอยู่ในช่วงระหว่าง 0 และ 1 โดยค่า 0 แสดงถึงการคัดแยกข้อมูลผิดพลาดทั้งหมดอย่างสมบูรณ์ และค่า 1 แสดงถึงการคัดแยกข้อมูลถูกต้องทั้งหมดอย่างสมบูรณ์

ตาราง 1. คุณลักษณะของข้อมูลจราจรทางคอมพิวเตอร์ที่เลือกใช้

ชื่อย่อ	คุณลักษณะ	รายละเอียด
f_1	Packet	จำนวนชุดข้อมูล
f_2	Byte	ผลรวมขนาดชุดข้อมูล
f_3	Session	จำนวนเซสชั่น
f_4	SrcAddr	จำนวนที่อยู่ต้นทาง
f_5	DstAddr	จำนวนที่อยู่ปลายทาง
f_6	SrcPort	จำนวนพอร์ตต้นทาง
f_7	DstPort	จำนวนพอร์ตปลายทาง
f_8	ΔAddr	ผลต่างจำนวนที่อยู่ต้นทางและปลายทาง
f_9	ΔPort	ผลต่างจำนวนพอร์ตต้นทางและปลายทาง



รูปที่ 2. ประสิทธิภาพการตรวจจับความผิดปกติในแต่ละประเภท

4. ผลการทดลองและการวิเคราะห์

ในงานวิจัยนี้ เราแบ่งขั้นตอนออกเป็นสองการทดลอง การทดลองที่หนึ่งเพื่อเปรียบเทียบประสิทธิภาพการตรวจจับความผิดปกติของแต่ละอัลกอริทึมและแต่ละคุณลักษณะของข้อมูลเมื่อทำงานร่วมกับโมเดลที่นำเสนอ การทดลองที่สองเพื่อจับเวลาที่ใช้ในการประมวลผลทั้งในขั้นตอนการเรียนรู้และในขั้นตอนการตรวจจับความผิดปกติ

4.1. การทดลองที่ 1: ประสิทธิภาพการตรวจจับ

การทดลองนี้เริ่มต้นด้วยการเลือกอัลกอริทึมและคุณลักษณะของข้อมูลทีละหนึ่ง แล้วป้อนชุดข้อมูลเรียนรู้เข้าไปในระบบ หลังจากนั้นจึงให้อัลกอริทึมดังกล่าวตรวจจับความผิดปกติในแต่ละชุดข้อมูลทดสอบ และวัดความถูกต้องแม่นยำในการตรวจจับความผิดปกติในแต่ละประเภท ได้ผลดังแสดงในรูปที่ 2 โดยแกน x แสดงถึงคุณลักษณะของข้อมูลจราจรทางคอมพิวเตอร์ที่เลือกใช้ เริ่มจาก f_1 จนถึง f_9 ดังรายละเอียดในตารางที่ 1 และ f_{all} หมายถึงการเลือกใช้คุณลักษณะทั้งหมดพร้อมกัน แกน y แสดงถึงค่า F-score ที่มีค่าสูงสุดเป็น 1 แสดงถึงความถูกต้องแม่นยำ 100% กราฟแท่งในรูปย่อยแสดงถึงประสิทธิภาพในการตรวจจับของแต่ละอัลกอริทึม ได้แก่ multivariate normal distribution (MND), k-nearest neighbor (KNN), และ one-class support vector machine (OSVM) ตามลำดับ ในรูปนี้มีรูปย่อยทั้งหมด 5 แถว แต่ละแถวแสดงถึงชุดข้อมูลทดสอบแต่ละประเภทของการใช้งานผิดปกติ ได้แก่ Back, IpSweep, Neptune, PortSweep, และ Smurf ตามลำดับจากบนลงล่าง

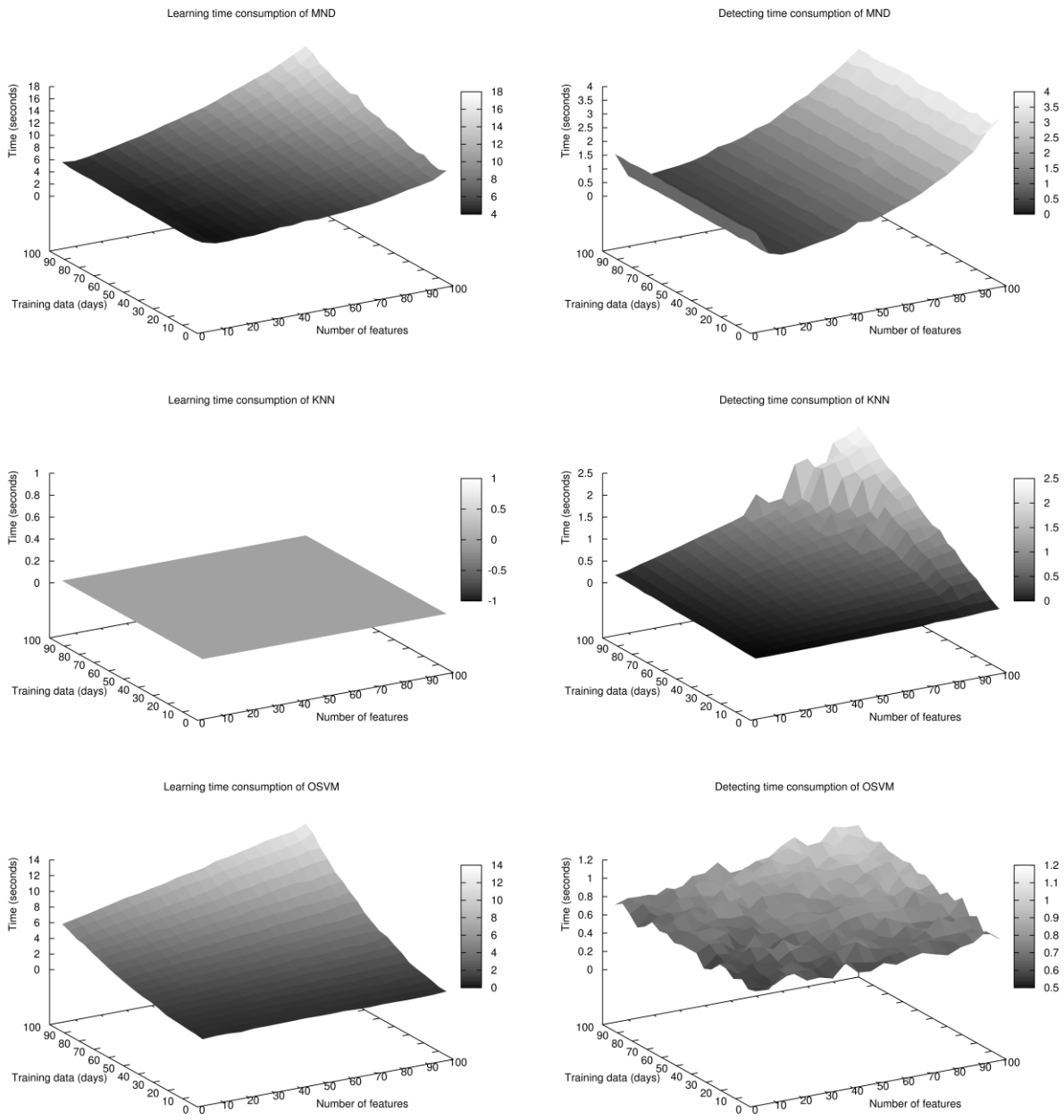
จากผลการทดลองที่ 1 เราสามารถกล่าวได้ว่า อัลกอริทึมที่มีประสิทธิภาพมากที่สุดสำหรับการตรวจจับความผิดปกติคือ KNN เพราะมีค่า F-score โดยรวมดีกว่าอัลกอริทึมอื่น โดยเฉพาะเมื่อตรวจจับความผิดปกติประเภท Back, Neptune, และ Smurf จะมีค่า F-score สูงเข้าใกล้หรือเท่ากับ 1 ซึ่งเป็นค่าสูงสุด อัลกอริทึมที่มีประสิทธิภาพรองลงมาคือ MND เพราะโดยรวมแล้วมีค่า F-score รองจาก KNN เพียงเล็กน้อย แต่อัลกอริทึม MND มีประสิทธิภาพด้อยที่สุดเมื่อใช้คุณลักษณะข้อมูล f1-f9 สำหรับการตรวจจับความผิดปกติประเภท IpSweep ส่วนอัลกอริทึมที่มีประสิทธิภาพด้อยที่สุดคือ OSVM เพราะมีค่า F-score โดยรวมแล้วอยู่ที่ประมาณ 0.25 หรือต่ำกว่า สาเหตุหลักเป็นเพราะอัลกอริทึม OSVM มีการตอบสนองไว (sensitivity) ต่อข้อมูลที่มีความแปรปรวนสูง ซึ่งปกติแล้วข้อมูลจราจรทางคอมพิวเตอร์จะมีความแปรปรวนสูงมาก จึงอาจเป็นสาเหตุที่ทำให้การตรวจจับความผิดปกติด้วยอัลกอริทึม OSVM มีประสิทธิภาพด้อยกว่าอัลกอริทึมอื่น

4.2. การทดลองที่ 2: เวลาในการประมวลผล

การทดลองนี้เราใช้ชุดข้อมูลเรียนรู้และชุดข้อมูลทดสอบเช่นเดียวกับการทดลองที่ 1 แต่เนื่องจากจุดประสงค์การทดลองคือการจับเวลาที่ใช้ในการประมวลผล ไม่สนใจประสิทธิภาพในการตรวจจับ ดังนั้นเราจึงสามารถนำชุดข้อมูลเรียนรู้และชุดข้อมูลทดสอบดังกล่าวมาจำลองทำซ้ำเพื่อเพิ่มวันและเพิ่มคุณลักษณะของข้อมูลให้ได้จำนวนตามที่เราต้องการ

การทดลองที่ 2 เริ่มต้นด้วยการจำลองชุดข้อมูลเรียนรู้จำนวน 5 วัน และมีคุณลักษณะของข้อมูลจำนวน 5 อย่าง แล้วป้อนชุดข้อมูลเรียนรู้ดังกล่าวเข้าไปในระบบ หลังจากนั้นจับเวลาที่ใช้ในการประมวลผลในขั้นตอนการเรียนรู้ แล้วค่อยๆเพิ่มจำนวนวันและจำนวนคุณลักษณะของข้อมูลครั้งละ 5 จนถึง 100 ส่วนในขั้นตอนการตรวจจับ ชุดข้อมูลทดสอบจะมีจำนวนคงที่ที่ 1 วัน แต่จะจำลองคุณลักษณะของข้อมูลเริ่มที่จำนวน 5 อย่างและเพิ่มครั้งละ 5 จนถึง 100 เช่นกัน แล้วจับเวลาที่ใช้ในการประมวลผลในขั้นตอนการตรวจจับ ได้ผลดังแสดงในรูปที่ 3 ซึ่งมีรูปย่อยทั้งหมด 6 รูป แบ่งออกได้เป็น 3 แถว แต่ละแถวแสดงถึงเวลาในการประมวลผลของแต่ละอัลกอริทึม ได้แก่ multivariate normal distribution (MND), k-nearest neighbor (KNN), และ one-class support vector machine (OSVM) ตามลำดับจากบนลงล่าง รูปด้านซ้ายมือแสดงผลจากขั้นตอนการเรียนรู้ ส่วนรูปด้านขวามือแสดงผลจากขั้นตอนการตรวจจับ ในแต่ละรูปย่อยมีรายละเอียดดังนี้ แกน x แสดงถึงจำนวนคุณลักษณะของข้อมูล (number of features) แกน y แสดงถึงจำนวนชุดข้อมูลเรียนรู้ (training data) มีหน่วยเป็นวัน (days) และแกน z แสดงถึงเวลาที่ใช้ในการประมวลผล (time) มีหน่วยเป็นวินาที (seconds)

จากผลการทดลองที่ 2 เราสามารถกล่าวได้ว่า สำหรับอัลกอริทึม MND ในขั้นตอนการเรียนรู้ใช้เวลาในการประมวลผลแปรผันโดยตรงกับจำนวนชุดข้อมูลเรียนรู้และจำนวนคุณลักษณะของข้อมูล ส่วนในขั้นตอนการตรวจจับใช้เวลาในการประมวลผลแปรผันโดยตรงกับจำนวนคุณลักษณะของข้อมูลเพียงอย่างเดียว สำหรับอัลกอริทึม KNN ในขั้นตอนการเรียนรู้ใช้เวลาในการประมวลผลเป็นค่าคงที่ ส่วนในขั้นตอนการตรวจจับใช้เวลาในการประมวลผลแปรผันโดยตรงกับจำนวนชุดข้อมูลเรียนรู้และจำนวนคุณลักษณะของข้อมูล สำหรับอัลกอริทึม OSVM ในขั้นตอนการเรียนรู้ใช้เวลาในการประมวลผลแปรผันโดยตรงกับจำนวนชุดข้อมูลเรียนรู้และจำนวนคุณลักษณะของข้อมูล ส่วนในขั้นตอนการตรวจจับใช้เวลาในการประมวลผลแบบสุ่ม (random) แต่มีแนวโน้มเพิ่มขึ้นตามจำนวนชุดข้อมูลเรียนรู้และจำนวนคุณลักษณะของข้อมูล ซึ่งเวลาในขั้นตอนนี้ขึ้นอยู่กับจำนวนเวกเตอร์และลักษณะของข้อมูลที่เกิดขึ้น จึงอาจเป็นสาเหตุหลักที่ทำให้ใช้เวลาในการประมวลผลแบบสุ่ม



รูปที่ 3. เวลาในการประมวลผลของแต่ละอัลกอริทึม (ซ้าย) จากขั้นตอนการเรียนรู้ (ขวา) จากขั้นตอนการตรวจจับ

เวลาในการประมวลผลที่ได้จากการทดลองยืนยันว่าโมเดลที่นำเสนอสามารถตรวจจับความผิดปกติแบบเรียลไทม์ได้ เราสามารถแบ่งการประมวลผลได้ออกเป็น 2 ขั้นตอนหลัก คือ ขั้นตอนการเรียนรู้และขั้นตอนการตรวจจับ การตรวจจับความผิดปกติแบบเรียลไทม์มีข้อจำกัดทางด้านเวลาในขั้นตอนการตรวจจับ เนื่องจากในขั้นตอนการตรวจจับเราต้องการใช้เวลาในการประมวลผลให้สั้นที่สุด ส่วนในขั้นตอนการเรียนรู้เราสามารถประมวลผลแบบออฟไลน์ได้ จากผลการทดลองแสดงให้เห็นว่าโมเดลที่นำเสนอใช้เวลาเพียงเล็กน้อยในการประมวลผลทั้งในขั้นตอนการเรียนรู้และในขั้นตอนการตรวจจับ ตัวอย่างเช่นจากผลการทดลองที่ได้จากอัลกอริทึม MND ในขั้นตอนการเรียนรู้ใช้เวลารวมทั้งสิ้นในการประมวลผลสูงสุดประมาณ 18 วินาที (รูปที่ 3 แถวบนซ้าย) แต่ในการทำงานจริง เราแบ่งการประมวลผลออกเป็นช่วงเวลาละ 1 วินาที ดังนั้นใน 1 วันมีการแบ่งเวลาออกเป็น 60 วินาที x 60 นาที x 24 ชั่วโมง ซึ่งเท่ากับ 86,400 ช่วงเวลา ดังนั้นเวลาที่แท้จริงในการประมวลผลต่อ 1 ช่วงเวลาเท่ากับ 18 / 86,400 ≈ 0.21 มิลลิวินาที ด้วยวิธีการคำนวณเช่นเดียวกันนี้ ในขั้นตอนการตรวจจับมีเวลารวมทั้งสิ้นในการประมวลผลสูงสุดประมาณ 4 วินาที (รูปที่ 3 แถวบนขวา) ดังนั้นเวลาที่แท้จริงในการประมวลผลต่อ 1 ช่วงเวลาเท่ากับ 4 / 86,400 ≈ 46.3 ไมโครวินาที ผลการทดลองที่ได้จากอัลกอริทึม KNN และ OSVM ก็เป็นแนวทางเดียวกัน ดังนั้นเราจึงสามารถสรุปได้ว่า เมื่อใช้โมเดลที่นำเสนอร่วมกับอัลกอริทึมดังกล่าว มีความเป็นไปได้สูงที่จะสามารถนำไปตรวจจับความผิดปกติแบบเรียลไทม์ในระบบเครือข่ายคอมพิวเตอร์ได้

5. สรุปและข้อเสนอแนะสำหรับการพัฒนาต่อไป

เป้าหมายของงานวิจัยนี้คือการพัฒนาวิธีการตรวจจับความผิดปกติในระบบเครือข่ายคอมพิวเตอร์ให้สามารถทำงานได้โดยอัตโนมัติแบบเรียลไทม์ ในบทความนี้จึงได้แสดงให้เห็นถึงปัญหาของวิธีการตรวจจับความผิดปกติที่มีอยู่ในปัจจุบัน โดยการจัดประเภทและวิเคราะห์โมเดลการเรียนรู้และการตรวจจับความผิดปกติ เพื่อแก้ปัญหาของโมเดลที่มีอยู่ในปัจจุบัน ผู้วิจัยจึงได้นำเสนอโมเดลรูปแบบใหม่ที่สามารถตรวจจับความผิดปกติได้ในแบบเรียลไทม์ และเพื่อทดสอบประสิทธิภาพการทำงานของโมเดลรูปแบบใหม่ที่นำเสนอ จึงได้ทำการทดลองเบื้องต้นเพื่อศึกษาความเป็นไปได้ของโมเดลดังกล่าวในการตรวจจับความผิดปกติแบบเรียลไทม์ในระบบเครือข่ายคอมพิวเตอร์

ในงานวิจัยนี้ ได้แบ่งการทดลองออกเป็น 2 ส่วน การทดลองส่วนที่หนึ่งเพื่อวัดประสิทธิภาพการตรวจจับความผิดปกติ ในการทดลองนี้เราได้เก็บรวบรวมข้อมูลจราจรทางคอมพิวเตอร์จากระบบเครือข่ายคอมพิวเตอร์ที่มีการใช้งานจริง และเลือกข้อมูลการใช้งานที่ผิดปกติรูปแบบต่างกันจำนวน 5 ประเภทจากชุดทดสอบ (test bed) นอกจากนี้เราได้เลือก 3 อัลกอริทึมให้มาทำงานร่วมกับโมเดลที่นำเสนอ ได้แก่ multivariate normal distribution, k-nearest neighbor, และ one-class support vector machine การทดลองส่วนที่สองเพื่อจับเวลาที่ใช้ในการประมวลผล ในการทดลองนี้เราได้จำลองข้อมูลเพื่อดูแนวโน้มของเวลาในการประมวลผลเมื่อมีจำนวนข้อมูลเพิ่มขึ้น ซึ่งเราได้จับเวลาในการประมวลผลทั้ง 2 ขั้นตอน คือ ขั้นตอนการเรียนรู้และขั้นตอนการตรวจจับความผิดปกติ

จากผลการทดลองแสดงให้เห็นว่า โมเดลที่นำเสนอมีความยืดหยุ่นในการทำงาน เพราะเราสามารถปรับแต่งการทำงานโดยการเลือกใช้อัลกอริทึมหรือคุณลักษณะของข้อมูลรูปแบบต่างๆได้ อย่างไรก็ดี การเลือกใช้อัลกอริทึมและคุณลักษณะของข้อมูลมีผลกระทบโดยตรงต่อประสิทธิภาพการตรวจจับความผิดปกติในระบบเครือข่ายคอมพิวเตอร์ นอกจากนี้ผลการทดลองยังชี้ให้เห็นว่า โมเดลที่นำเสนอใช้เวลาประมวลผลที่รวดเร็วทั้งในขั้นตอนการเรียนรู้และในขั้นตอนการตรวจจับความผิดปกติ และมีความเป็นไปได้สูงที่โมเดลดังกล่าวสามารถนำไปประยุกต์ใช้ทำงานแบบเรียลไทม์ได้

การทดลองในบทความนี้เป็นเพียงการทดลองเบื้องต้นเพื่อดูความเป็นไปได้ในการนำโมเดลดังกล่าวไปใช้งาน ยังต้องมีการทดสอบความสามารถในอีกหลายส่วน อาทิเช่น ทดสอบการเรียนรู้จากชุดข้อมูลที่มีข้อมูลการใช้งานที่ปกติและข้อมูลการใช้งานที่ผิดปกติปะปนกัน ทดสอบกับความผิดปกติรูปแบบอื่น หรือแม้กระทั่งทดสอบประสิทธิภาพการตรวจจับความผิดปกติเมื่อมีจำนวนชุดข้อมูลเรียนรู้ที่ต่างกัน และท้ายที่สุดถูกพัฒนานำไปใช้งานจริงกับระบบเครือข่ายคอมพิวเตอร์

6. กิตติกรรมประกาศ

ขอขอบคุณผู้ดูแลระบบศูนย์บริการคอมพิวเตอร์และอินเตอร์เน็ต มหาวิทยาลัยเกษตรศาสตร์ ที่ช่วยอำนวยความสะดวกในการเก็บข้อมูลจราจรคอมพิวเตอร์ งานวิจัยนี้ได้รับทุนสนับสนุนจากโครงการพัฒนาอาจารย์ของมหาวิทยาลัยกรุงเทพ (Faculty Development Program)

เอกสารอ้างอิง

[1] A. Patcha and J.M. Park, “An overview of anomaly detection techniques: Existing solutions and latest technological trends,” *Computer Networks*, vol. 51, no. 12, pp. 3448-3470, Aug. 2007.

[2] V. Chandola, A. Banerjee, V. Kumar, “Anomaly detection: A survey,” *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1-15:58 , Jul. 2009.

[3] M. Roesch, “Snort lightweight intrusion detection for networks,” *in Proceedings of the 13th USENIX conference on System administration*, USENIX Association, pp. 229-238, 1999.

[4] S. Jiang, X. Song, H. Wang, J.J. Han, Q.H. Li, “A clustering-based method for unsupervised intrusion detections,” *Pattern Recognition Letter*, vol. 27, no. 7, pp. 802-810, May 2006.

[5] R.P. Lippmann, D.J. Fried, I. Graf, J.W. Haines, “Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation,” *IDSCEX00 Proceedings*, vol. 2, pp. 12-26.

[6] S. Theodoridis, K. Koutroumbas, “Pattern Recognition Fourth Edition,” Academic Press, 2008.

[7] Thomas M. Mitchell, “Machine Learning First Edition,” McGraw-Hill, Inc., New York, NY, USA., 1997.

[8] B. Scholkopf, J.C. Shawe-Taylor, A.J. Smola, R.C. Williamson, “Estimating the support of a high-dimensional distribution,” *Neural Computation*, vol. 13, no. 7, pp. 1443-1471, 2001.

[9] C.C. Chang, C.J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1-27:27, 2011.

[10] P. Baldi, S. Brunak, Y. Chauvin, C. Andersen, H. Nielsen, “Assessing the accuracy of prediction algorithms for classification: an overview,” *Bioinformatics*, vol. 16, no. 5, pp. 412-424, 2007.